

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Menurut Jerry FithGerald, sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan dan berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.(Parno, 2010,153)

Menurut Ludwig Von Bartalanfy, sistem merupakan seperangkat unsur yang saling terikat dalam suatu antar relasi di antara unsur-unsur tersebut dengan lingkungan.(Hasbon Hendra, 2012,83)

Menurut Jogiyanto Hartono dalam buku yang berjudul Analisis dan Desain Sistem Informasi menjelaskan bahwa “Sistem adalah sekumpulan dari elemen-elemen yang berinteraksi untuk mencapai tujuan tertentu”. (Jurnal Riset Komputerisasi Akuntansi @is The Best, Vol.1, No.1, April 2012 : Hal, 41-60)

II.1.1. Karakteristik

Suatu sistem mempunyai karakteristik sebagai berikut:

a. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen

sistem dapat berupa suatu subsistem atau bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya selalu mengandung komponen-komponen.

b. Batas Sistem (*Boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan.

c. Lingkungan Luar Sistem (*Environment*)

Lingkungan luar suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara.

d. Penghubung Sistem (*Interface*)

Penghubung merupakan media penghubung antara satu subsistem dengan yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya. Keluaran dari satu sistem akan menjadi masukan untuk subsistem yang lainnya dengan melalui penghubung.

e. Masukan Sistem (*Input*)

Masukan adalah energi yang dimasukkan ke dalam sistem yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*) dapat. Sinyal input adalah energi yang diproses untuk didapat keluaran.

f. Keluaran Sistem (*Output*)

Keluaran adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem.

g. Pengolahan Sistem (*Process*)

Suatu sistem dapat mempunyai suatu proses yang akan mngubah masukan menjadi keluaran. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

h. Sasaran Sistem (*Objective*)

Suatu sistem pasti mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.(Tata Sutabri,2012,20) .

II.1.2. Metode Sistem

a. Blackbox Approach

Suatu sistem di mana input dan outputnya dapat didefinisikan tetapi prosesnya tidak diketahui atau tidak didefinisi. Metode ini hanya dapat dimengerti oleh pihak dalam (yang menangani) sedangkan pihak luar hanya mengetahui masukan dan hasilnya. Sistem ini terdapat pada subsistem tingkat rendah. Contohnya : bagian percetakan uang, proses pencernaan. (Asbon Hendra : 2012 : 166)

b. Analytic System

Suatu metode yang mencoba untuk melihat hubungan seluruh masalah untuk menyelidiki kesistematiskan tujuan dari sistem yang tidak efektif dan evaluasi pilihan dalam bentuk ketidakefektifan dan biaya. Dalam metode ini, beberapa langkah diberikan seperti di bawah ini :

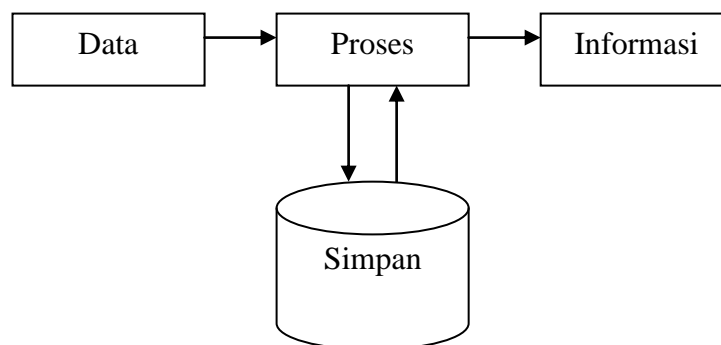
- a. Menentukan identitas dari sistem.
 1. Sistem apa yang diterapkan.
 2. Batasannya.
 3. Apa yang dilaksanakan sistem tersebut.
- b. Menentukan tujuan dari sistem.
 1. Output yang dihasilkan dari isi sistem.
 2. Fungsi dan tujuan yang diminta untuk mencoba menanggulangi lingkungan.
- c. Bagian-bagian apa saja yang terdapat dalam sistem dan apa tujuan dari masing-masing bagian tersebut.
 1. Tujuan masing-masing bagian harus jelas.
 2. Cara apa yang digunakan subsistem untuk berhubungan dengan subsistem lain.
- d. Bagaimana bagian-bagian yang ada dalam sistem itu saling berhubungan menjadi satu kesatuan. (Asbon Hendra : 2012 : 167)

II.2. Informasi

Dalam buku pengantar Sistem Informasi menjelaskan bahwa “ Informasi merupakan data yang diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat.(Asbon Hendra : 2012: 167)

Menurut Krismiaji Informasi adalah data yang diorganisasi dan telah memiliki kegunaan dan manfaat. (Jurnal Riset Komputerisasi Akuntansi @is The Best, Vol.1, No.1, April 2012 : Hal, 41-60)

Jadi informasi dapat didefinisikan sebagai data yang telah diolah menjadi suatu bentuk yang berarti bagi penerima dan bermakna dalam pengambilan keputusan.



Gambar II.1. Pemrosesan Data

Sumber : Tata Sutabri (2012:2)

II.3. Sistem Informasi

Menurut Azhar Susanto Sistem Informasi adalah kumpulan dari sub-sub sistem baik fisik maupun non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai tujuan yaitu mengolah data menjadi informasi yang berguna. (Jurnal Riset Komputerisasi Akuntansi @is The Best, Vol.1, No.1, April 2012 : Hal, 41-60)

Robert A. Leitch menjelaskan bahwa “Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan”. (Asbon Hendra : 2012 : 169). Ada empat operasi dasar dari sistem informasi, yaitu mengumpulkan, mengelolah, menyimpan, dan menyebarkan informasi. Informasi mungkin dikumpulkan dari lingkungan dalam atau luar dan memungkinkan didistribusikan ke dalam atau keluar organisasi. (Asbon Hendra : 2012 : 169)

II.4. Akuntansi

Kegiatan selama periode akuntansi adalah kegiatan mencatat transaksi transaksi hingga kegiatan menutup buku, yang dapat dirinci sebagai berikut:

1. Jurnal yaitu: kegiatan mencatat transaksi-transaksi keuangan yang terjadi pada perusahaan.
2. Posting yaitu : kegiatan pembukuan catatan dari jurnal kedalam rekening buku besar yang bersangkutan.
3. Neraca saldo(trial balance) yaitu : kegiatan menguji kebenaran saldo-saldo debit dan kredit rekening buku besar dengan cara menyusun saldo-saldo rekening buku besar kedalam suatu daftar Yng disebut dengan neraca saldo.
4. Ayat Penyesuaian(adjusting entries) yaitu: kegiatan penyesuaian jumlah-jumlah tersebut sesuai dengan keadaan yang sebenarnya pada akhir periode.

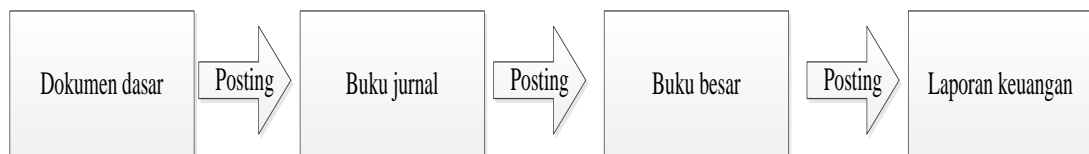
5. Laporan Keuangan (*financial statement*) yaitu : kegiatan menyusun neraca(*balance sheet*), laporan laba rugi(*income statement*), dan laporan sisalaba berdasarkan data-data neraca saldo yang telah disesuaikan.
6. Ayat penutup(*closing entries*) yaitu : kegiatan menyusun pos-pos penutup.

Akuntansi adalah sebuah sistem informasi yang menghasilkan keuangan kepada pihak-pihak yang berkepentingan mengenai aktivitas ekonomi dan kondisi suatu perusahaan.(Rudianto ; 2009 : 4).

Akuntansi adalah aktivitas mengumpulkan, menganalisis, menyajikan dalam bentuk angka, mengklasifikasi, mencatat, meringkas dan melaporkan aktivitas/transaksi perusahaan dalam bentuk informasi keuangan.(Rudianto ; 2009 : 14)

II.4.1. Siklus Akuntansi

Siklus akuntansi adalah urutan kerja yang harus dibuat oleh akuntan, sejak awal hingga menghasilkan laporan keuangan suatu perusahaan.



Gambar II.2. Siklus Akuntansi

Sumber : Rudianto (2009:14)

Keterangan gambar:

- a. Dokumen dasar : Bukti transaksi yang dijadikan dasar oleh akuntan untuk mencatat, seperti : faktur, kuitansi, nota penjualan, invoice, dll.
- b. Jurnal (*Journal*) : Aktivitas meringkas dan mencatat transaksi perusahaan berdasarkan dokumen besar. Tempat untuk mencatat transaksi tersebut disebut Buku Jurnal.
- c. Posting : Aktivitas memindahkan catatan di buku jurnal kedalam buku besar sesuai jenis transaksi dan nama perkiraan masing-masing.
- d. Buku besar (*General ledger*) : Kumpulan dari semua akun/ perkiraan yang diperoleh dimiliki oleh perusahaan yang saling berhubungan satu sama lainnya dan berupa satu kesatuan.
- e. Akun/ perkiraan (*Account*) : Suatu kelas informasi didalam suatu sistem akuntansi. Atau suatu media yang digunakan untuk mencatat informasi sumber daya perusahaan dan informasi lainnya berdasarkan jenisnya. Misalnya perkiraan kas, perkiraan piutang, akun modal, dsb.(Rudianto ; 2009 : 14).

II.5. Sistem Informasi Akuntansi

II.5.1. Sistem Informasi Akuntansi

Definisi sistem informasi akuntansi menurut Krismiaji dalam bukunya yang berjudul Sistem Informasi Akuntansi adalah sebuah sistem yang memproses data transaksi guna menghasilkan informasi yang bermanfaat untuk merencanakan, mengendalikan dan mengoperasikan bisnis. (Jurnal Riset Komputerisasi Akuntansi @is The Best, Vol.1, April 2012 : Hal, 41-60)

Sistem Informasi Akuntansi (SIA) merupakan suatu rerangka pengkoordinasian sumber daya(*data, materials, equipment, suppliers, personal, and funds*) untuk mengkonversi input berupa data ekonomik menjadi keluaran berupa informasi keuangan yang digunakan untuk melaksanakan kegiatan suatu entitas dan menyediakan informasi akuntansi bagi pihak-pihak yang berkepentingan(Wilkinson,1991,45).

Transaksi memungkinkan perusahaan melakukan operasi, menyelenggarakan arsip dan catatan yang *up to date*, dan mencerminkan aktivitas organisasi. Tipe transaksi dasar adalah:

1. Penjualan produk atau jasa
2. Pembelian bahan baku, barang dagangan, jasa, dan aset tetap dari supplier,
3. Penerimaan kas
4. Pengeluaran kas
5. Pengeluaran kas gaji karyawan.(Tata Sutabri,2012,83)

Sebagai pengolah transaksi, sistem informasi akuntansi berperan mengatur dan mengoperasionalkan semua aktivitas transaksi perusahaan. Adapun tujuan sistem

informasi akuntansi adalah untuk menyediakan informasi yang diperlukan dalam pengambilan keputusan yang dilaksanakan oleh aktivitas yang disebut pemrosesan informasi. Pengguna utama pemrosesan transaksi adalah manajer perusahaan. Pengguna output lainnya adalah para karyawan penting seperti akuntan, insinyur serta pihak luar seperti investor dan kreditor. (Tata Sutabri, 2012, 84) .

II.6. Aktiva Tetap

Aktiva tetap adalah barang berwujud milik perusahaan yang sifatnya relative permanen dan digunakan dalam kegiatan normal perusahaan, bukan untuk diperjualbelikan. (Rudianto ; 2009 : 272)

Menurut Baridwan (2004:27), yang dimaksud dengan asset jangka panjang dalam hal ini asset tetap adalah asset-aset yang berwujud yang sifatnya relative permanen yang digunakan dalam kegiatan perusahaan yang normal. Istilah relative permanen menunjukkan sifat aktif yang bersangkutan dapat digunakan dalam jangka waktu yang relative cukup lama. Untuk tujuan akuntansi, jangka waktu penggunaan dibatasi dengan “lebih dari satu periode akuntansi”. Jadi asset berwujud yang umurnya lebih dari satu periode akuntansi dikelompokkan sebagai asset tetap berwujud. (Jurnal akuntansi dan keuangan Volume 16 Nomor 2, Juli-Desember 2011)

Untuk dapat dikelompokkan sebagai aktiva tetap, maka suatu aktiva harus memiliki kriteria sebagai tertentu, yaitu:

1. Berwujud
2. Umurnya lebih dari satu tahun

3. Digunakan dalam operasi perusahaan
4. Tidak diperjualbelikan
5. Material
6. Dimiliki perusahaan.(Rudianto ; 2009 : 273).

II.7. Penyusutan

Penyusutan adalah pengalokasian harga perolehan aktiva tetap menjadi beban kedalam periode akuntansi yang menikmati manfaat dari aktiva tetap tersebut.(Rudianto ; 2009 : 276).

II.7.1. Metode Perhitungan Depresiasi

Untuk mengalokasikan harga perolehan suatu aktiva tetap ke dalam periode-periode yang menikmati aktiva tetap tersebut, bukan hanya menggunakan suatu metode saja. Terdapat beberapa metode yang dapat digunakan untuk menghitung beban depresiasi periodik, yaitu :

1. Metode Garis Lurus (Straight Line Method)

Adalah suatu metode penghitungan depresiasi aktive tetap di mana setiap periode akuntansi diberikan beban yang sama secara merata. Beban depresiasi dihitung dengan cara mengurangi harga perolehan dengan nilai sisa dan dibagi dengan umur ekonomis dari aktiva tetap tersebut (Rudianto : 2009 : 277).

$$Depresiasi = \frac{\text{Harga Perolehan} - \text{Nilai Sisa}}{\text{Taksiran Umur Ekonomis Aktiva}}$$

Metode perhitungan depresiasi dengan metode garis lurus akan menghasilkan beban depresiasi aktiva tetap yang sama dari tahun ke tahun. Metode ini juga dapat menghasilkan beban depresiasi berupa suatu persentase dari harga perolehan aktiva tetap tersebut.

2. Metode jam jasa (Service Hour Method)

Adalah suatu metode penghitungan depresiasi aktiva tetap, di mana beban depresiasi pada suatu periode akuntansi dihitung berdasarkan beberapa jam periode akuntansi tersebut mempergunakan aktiva tetap itu. Semakin lama aktiva tetap tersebut dipergunakan di dalam suatu periode, akan semakin besar pula beban depresiasinya. Demikian pula sebaliknya. Besarnya beban depresiasi aktiva tetap dihitung dengan cara mengurangkan taksiran nilai residu dari harga perolehannya dan membagi hasilnya dengan taksiran jumlah jam pemakaian total dari aktiva tetap tersebut sepanjang umur ekonomisnya. Hasil dari pembagian tersebut adalah beban depresiasi per jam. Jumlah tersebut dijadikan dasar untuk mengalihkan dengan jumlah jam actual pemakaian aktiva tetap tersebut di dalam suatu periode, sehingga diketahui beban depresiasi aktiva tetap pada suatu periode. (Rudianto : 2009 : 278)

$$Depresiasi = \frac{\text{Harga Perolehan} - \text{Nilai Sisa}}{\text{Taksiran Jam Pemakaian Total}}$$

Beban depresiasi aktiva tetap yang dihitung dengan metode jam jasa akan menghasilkan tarif depresiasi per jam atau per satuan waktu tertentu. Dan berdasarkan tariff depresiasi tersebut, beban depresiasi suatu periode dihitung

dengan mengalikan tarif tersebut dengan jumlah jam atau waktu yang digunakan di dalam periode tersebut.

3. Metode Hasil Produksi (Productive Output Method)

Adalah suatu perhitungan depresiasi aktiva tetap, di mana beban depresiasi pada suatu periode akuntansi dihitung berdasarkan beberapa banyak produk yang dihasilkan periode akuntansi tersebut dengan mempergunakan aktiva tetap itu. Semakin banyak produk yang dihasilkan di dalam suatu periode, akan semakin besar pula beban depresiasinya. Demikian pula sebaliknya. Besarnya beban depresiasi aktiva tetap dihitung dengan cara mengurangkan taksiran nilai residu dari harga perolehannya dan membagi hasilnya dengan taksiran jumlah produk yang akan dihasilkan dari aktiva tetap tersebut sepanjang umur ekonomisnya. Hasil dari pembagian tersebut akan diketahui beban depresiasi per unit produk. Jumlah tersebut dijadikan dasar untuk mengalihkan dengan jumlah unit produk dihasilkan secara actual di dalam suatu periode, sehingga diketahui beban depresiasi aktiva tetap pada suatu periode. (Rudianto : 2009 : 278)

$$Depresiasi = \frac{\text{Harga Perolehan}-\text{Nilai Sisa}}{\text{Taksiran Jumlah Total yang Dapat Dihasilkan}}$$

Beban depresiasi aktiva tetap yang dihitung dengan metode hasil produksi akan menghasilkan tarif depresiasi per unit atau satuan tertentu. Dan berdasarkan tarif depresiasi tersebut, beban depresiasi suatu periode dihitung dengan mengalikan tarif tersebut dengan jumlah unit atau satuan lain yang digunakan di dalam periode tersebut.

II.7.2. Faktor yang mempengaruhi

Terdapat tiga faktor yang harus dipertimbangkan dalam menentukan beban penyusutan setiap periode, yaitu:

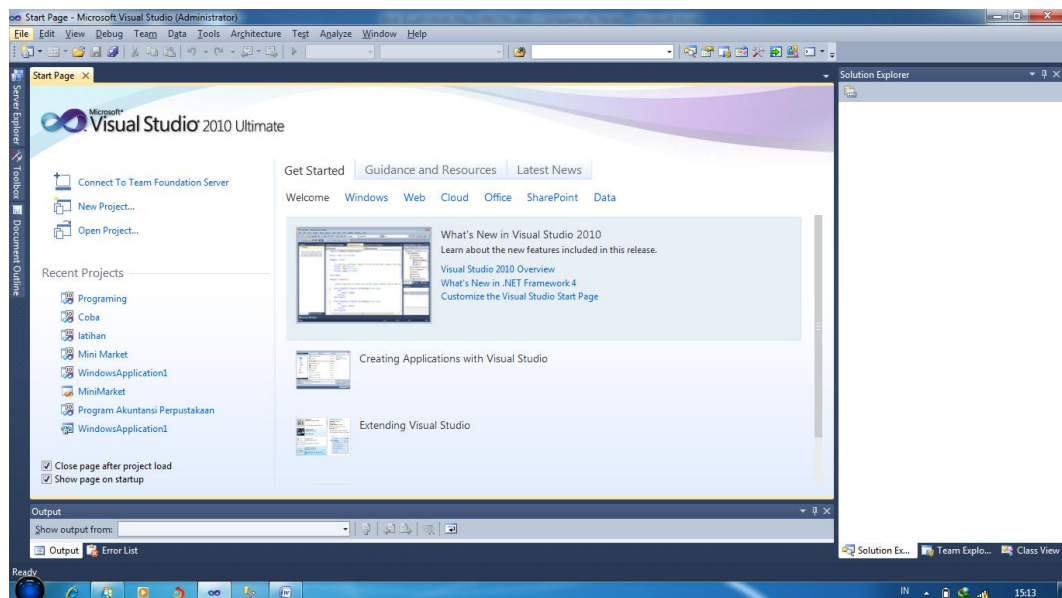
1. Harga perolehan adalah keseluruhan uang yang dikeluarkan untuk memperoleh suatu aktiva sampai siap digunakan oleh perusahaan.
2. Nilai sisa(*Residu*) adalah taksiran harga jual aktiva tetap tersebut pada akhir masa manfaat aktiva tetap tersebut. Jumlah taksiran nilai *residu* juga akan sangat dipengaruhi umur ekonomisnya, *inflasi*, nilai tukar mata uang, bidang usaha dan sebagainya.
3. Taksiran umur kegunaan adalah taksiran masa manfaat dari aktiva tetap tersebut. Masa manfaat adalah taksiran umur ekonomis dari aktiva tetap tersebut, bukan umur teknis. Taksiran masa manfaat dapat dinyatakan dalam satuan periode waktu, satuan hasil produksi atau satuan masa jam kerja.(Rudianto ; 2009 : 276).

II.8. Visual Basic 2010

Visual Basic 2010 merupakan produk pemrograman andalan dari Microsoft Corporation, dimana di dalamnya berisi beberapa jenis IDE pemrograman seperti Visual Basic, Visual ++, Visual Web Developer, Visual C#, dan Visual F#. Adapun database standar yang disertakan adalah Microsoft SQL Server 2008 express. Visual Basic 2010 merupakan versi perbaikan dan pengembangan dari versi pendahulunya, yaitu Visual Basic 2008 dan didukung perluasan terhadap database-database, baik standalone maupun database server. Didalam

mengkoneksikan Visual Basic 2010 dan Sql Server 2008 adalah dengan memanggil nama database yang telah dibuat didalam Sql Server baik dalam module maupun class.(Wahana Komputer,2011,3).

Bahasa Visual Basic 2010 awalnya berasal dari bahasa pemrograman yaitu bahasa BASIC, yang oleh Microsoft diadaptasi dalam program Microsoft Quick BASIC. Saat ini versi Microsoft Visual Studio yang beredar adalah versi 10 yang populer dengan nama Microsoft Visual Studio 2010.(Wahana Komputer,2011,3).

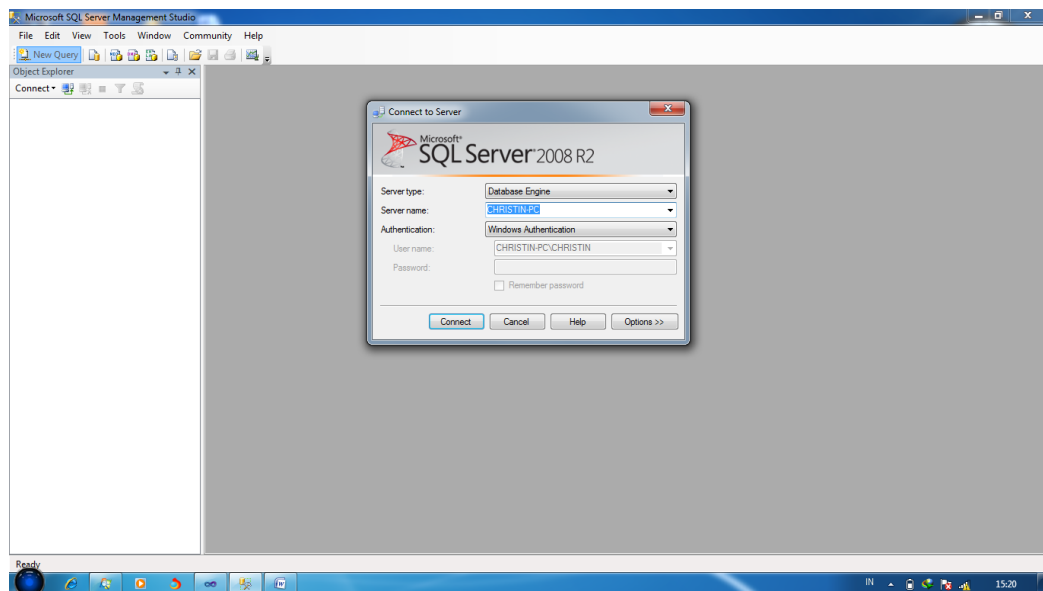


Gambar II.3. Microsoft Visual Basic 2010

II.9. SQL Server 2008

SQL Server 2008 adalah sebuah RDBMS (Relational Database Management System) yang di-develop oleh Microsoft. SQL Server 2008 menggunakan SQL language (*Structur Query Language*).Sebuah database berisi satu table atau lebih dan memiliki nama yang berbeda untuk masing-masing table, memiliki field-field

dan berisi record. Query digunakan untuk menyimpan dan mengolah data. Pada SQL Server 2008, kita bisa melakukan pengambilan dan modifikasi data yang ada dengan cepat dan efisien. Pada SQL Server 2008, kita bisa membuat object-object yang sering digunakan pada aplikasi bisnis seperti membuat database, table, function, store procedure, trigger, dan view. (Cybertron Solution dan SmitDev Community, 2010, 101)



Gambar II.4. Connect ke SQL Server 2008



Terdapat beberapa view yang akan sering digunakan pada SQL Server Management Studio, di antaranya Object Explorer yang digunakan untuk melakukan aktivitas pada database menggunakan GUI dan Query yang digunakan untuk melakukan aktivitas database menggunakan T-SQL Query. Pada SQL Server 2008, database adalah object yang paling vital karena pada object tersebutlah object dan data didefinisikan dan disimpan. (Cybertron Solution dan SmitDev Community, 2010, 103)

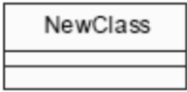



II.10. UML (Unified Modelling Language)



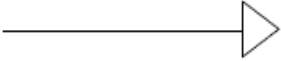
UML (Unified Modelling Language) adalah bahasa yang digunakan untuk memvisualisasikan, mendefinisikan, membangun dan membuat dokumen dari arsitektur perangkat lunak. UML dapat digunakan pada semua proses melalui metodologi pengembangan perangkat lunak dan melakukan implementasinya pada teknologi yang berbeda. (Jurnal TELEMATIKA MKOM, Vol.3 No.2, September 2011). Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (sharing) dan mengkomunikasikan rancangan mereka dengan yang lain.

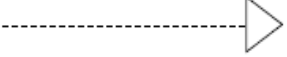
Dalam merancang UML (*Unified Modelling Language*) ada notasi-notasi yang telah ditentukan. Notasi-notasi dalam UML terdapat pada table II.1

Tabel II.1. Daftar Notasi UML

Simbol	Arti	Fungsi
	Notasi Actor	Actor menggambarkan segala pengguna software aplikasi (user).
	Notasi Use Case	Use case menjelaskan urutan kegiatan yang dilakukan actor dan system untuk mencapai suatu tujuan

		tertentu.
	Notasi Class	Class merupakan pembentuk utama dari system berorientasi obyek, karena class menunjukkan kumpulan obyek yang memiliki atribut dan operasi yang sama.
	Notasi Interface	Interface merupakan kumpulan operasi tanpa implementasi dari suatu Class.
	Notasi Interaction	Interaction digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar obyek.
	Notasi Note	Note digunakan untuk memberikan keterangan atau komentar tambahan dari suatu elemen sehingga bisa

		langsung terlampir dalam model.
	Notasi Dependency	Dependency merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen member pengaruh pada elemen lain.
	Notasi Asociation	Association menggambarkan navigasi antar class (navigation), berapa banyak obyek lain yang bias berhubungan dengan satu obyek (multiplicity antar class) dan apakah suatu class menjadi bagian dari class lainnya (aggregation).
	Notasi Generalization	Generalization menunjukkan hubungan antara elemen yang lebih umum ke elemen yang lebih spesifik.
	Notasi	Realization menunjukkan hubungan bahwa elemen yang ada di bagian tanpa panah akan merealisasikan apa

	Realization	yang dinyatakan oleh elemen yang ada di bagian dengan panah.
---	-------------	--

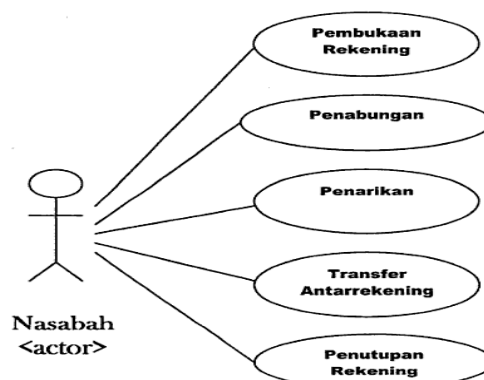
Tabel II.1 : Simbol - Simbol Notasi UML (Unified Modelling Language)

Sumber : Munawar (2005:23)

Dalam metode UML terdapat beberapa diagram sebagai berikut:

1. Use Case Diagram

Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. Use case diagram terdiri atas diagram untuk use case dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. Use case merepresentasikan operasi-operasi yang dilakukan oleh *actor*. Use case digambarkan berbentuk elips dengan nama operasi dituliskan di dalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke use case. Dibawah ini salah satu contoh diagram use case pada mesin ATM.



Gambar.II.5.Use Case Diagram

Sumber : (Adi Nugroho: 2009)

2. Class Diagram

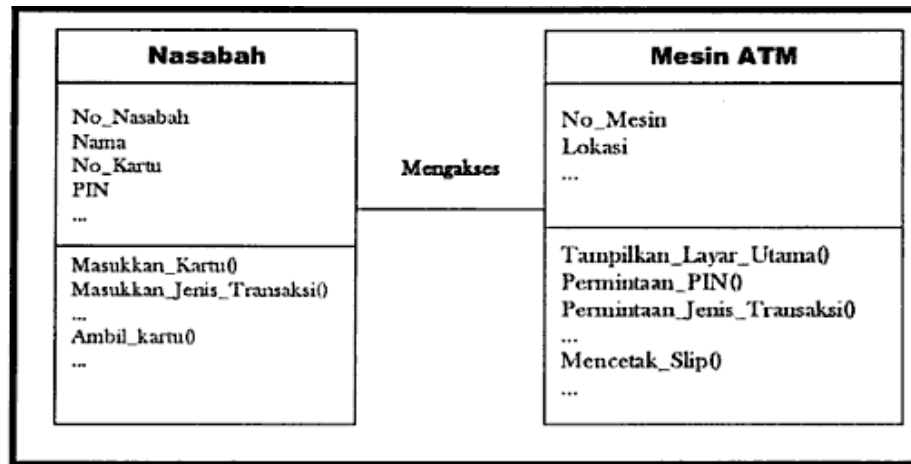
Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. Class diagram menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

- a. Private, tidak dapat dipanggil dari luar class yang bersangkutan
- b. Protected, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
- c. Public, dapat dipanggil oleh siapa saja.

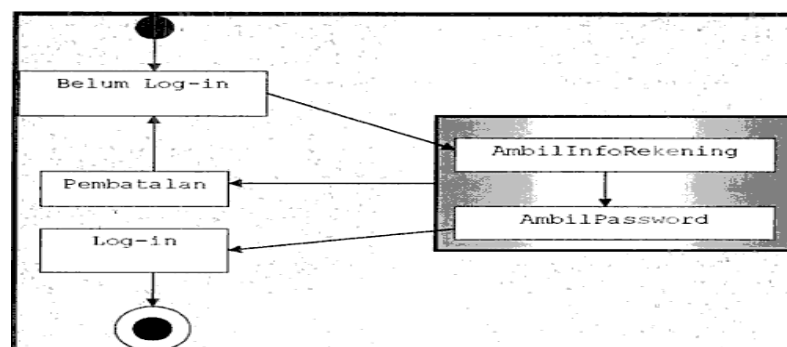


Gambar.II.6. Class Diagram

Sumber : (Adi Nugroho: 2009)

3. Statechart Diagram

Menggambarakan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya statechart diagram menggambarkan class tertentu (satu class dapat memiliki lebih dari satu statechart diagram).

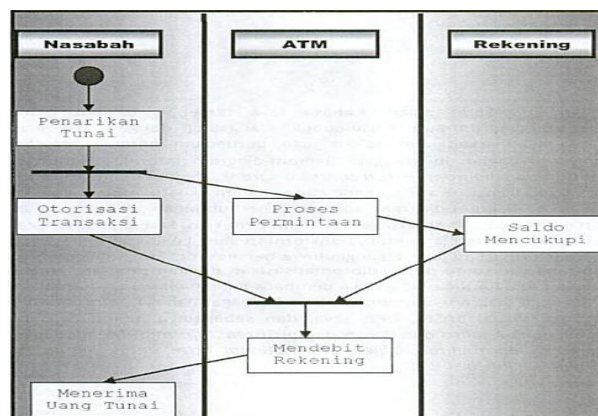


Gambar.II.7. Statechart Diagram

Sumber : (Adi Nugroho: 2009)

4. Activity Diagram

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Dibawah ini salah satu contoh diagram Aktiviti pada mesin ATM.



Gambar.II.8 Activity Diagram

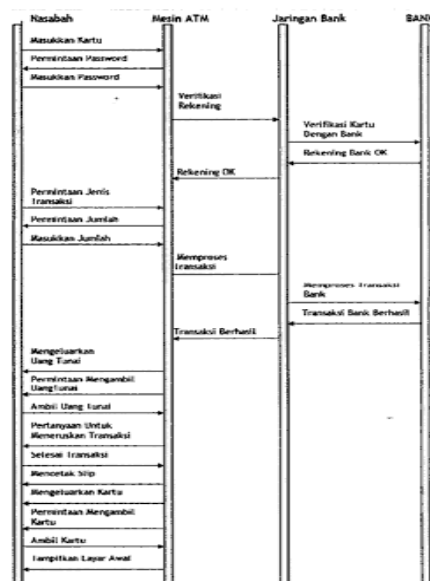
Sumber : (Adi Nugroho: 2009)

5. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram*

biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang *trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses.

Contoh *sequence diagram* :

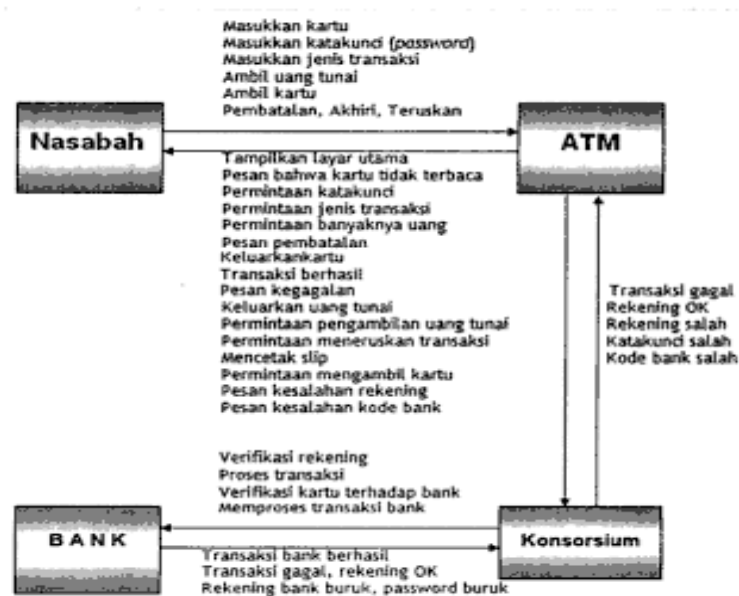


Gambar.II.9. Sequence Diagram

Sumber : (Adi Nugroho: 2009)

6. Collaboration Diagram

Collaboration Diagram adalah suatu diagram yang memperlihatkan/menampilkan pengorganisasian interaksi yang terdapat disekitar objek (seperti halnya *sequence diagram*) dan hubungannya terhadap yang lainnya. Collaboration Diagram lebih menekankan kepada peran setiap objek dan bukan pada waktu penyampaian pesan/*message*. Dibawah ini salah satu contoh diagram Aktiviti pada mesin ATM.



Gambar.II.10. Collaboration Diagram

Sumber : (Adi Nugroho: 2009)

7. Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link*

time, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa interface, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

8. Deployment/physical diagram

Menggambarkan detail bagaimana komponen di *deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisik. Sebuah node adalah server, workstation, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar node (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.

II.1.1 Database

Sebuah sistem informasi tidak lepas dari peran database. Para pengguna sistem informasi, seperti manajer, pelanggan, atau siapa saja bisa memperoleh informasi untuk berbagai kepentingan berkat keberadaan database. Data yang tersimpan dalam database dapat diambil dan diproses sehingga menghasilkan informasi.

Dalam bukunya Ramakrishnan dan Gerke,2000 menyatakan bahwa database adalah kumpulan data yang umumnya menjabarkan aktivitas-aktivitas dari satu atau lebih dari satu organisasi yang terkait.(Abdul Kadir,2009,9)

Jadi, sebuah database mencatat berbagai data yang diperlukan oleh suatu organisasi. Rekaman-rekaman data tersebut pada suatu saat akan diambil dan

melalui suatu pemrosesan akan diperoleh informasi yang dikehendaki oleh pengguna.

Data dalam sebuah database disusun berdasarkan sistem hierarki yaitu:

1. Database, merupakan kumpulan file yang saling terkait satu sama lain, misalnya file data induk karyawan, file jabatan, file penggajian dan lainnya.
2. File, yaitu kumpulan dari record yang saling terkait dan memiliki format field yang sama dan sejenis.
3. Record, yaitu kumpulan field yang menggambarkan suatu unit data individu tertentu.
4. Field, yaitu atribut dari record yang menunjukkan suatu item dari data, seperti nama, alamat, dan lain sebagainya.
5. Byte, yaitu atribut dari field yang berupa huruf yang membentuk nilai dari sebuah field. Huruf tersebut dapat berupa numerik maupun abjad atau karakter khusus.
6. Bit, yaitu bagian terkecil dari data keseluruhan, yaitu berupa karakter ASCII nol atau satu yang merupakan komponen pembentuk byte (Budi Sutedjo Dharma Oetomo:2006:102)

II.10.1. *Entity Relationship Diagram - ERD*

Pada perancangan konseptual diperlukan suatu pendekatan yang digunakan untuk menggambarkan hubungan antar data. Hubungan tersebut dapat dinyatakan dalam model bentuk E-R. Model ini adalah dasar yang penting dalam merancang database. Model E-R adalah suatu model yang digunakan untuk menggambarkan

data dalam bentuk entitas, atribut, dan hubungan antarentitas. Namun, model ini tidak mencerminkan bentuk fisik yang nantinya akan disimpan dalam database melainkan hanya bersifat konseptual.(Abdul Kadir,2009,30)

Entity Relationship(ER) data model didasarkan pada persepsi terhadap dunia nyata yang tersusun atas kumpulan objek-objek dasar yang disebut entitas dan hubungan antar objek. Entitas adalah sesuatu atau objek dalam dunia nyata yang dapat dibedakan dari objek lain. Sebagai contoh, masing-masing mahasiswa adalah entitas dan mata kuliah dapat pula dianggap sebagai entitas.

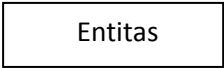
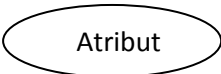


Entitas digambarkan dalam basis data dengan kumpulan atribut. Misalnya atribut NIM, nama, alamat, dan kota bisa menggambarkan data mahasiswa tertentu dalam suatu universitas. Atribut-atribut membentuk entitas mahasiswa. Demikian pula atribut kodeMK, namaMK, dan SKS mendeskripsikan entitas mata kuliah.

Atribut NIM digunakan untuk mengidentifikasi mahasiswa secara unik karena dimungkinkan terdapat dua mahasiswa dengan nama, alamat, dan kota yang sama. Pengenal unik harus diberikan pada masing-masing mahasiswa.

Relasi adalah hubungan antara beberapa entitas. Sebagai contoh, relasi menghubungkan mahasiswa dengan mata kuliah yang di ambilnya. Kumpulan semua entitas bertipe sama disebut kumpulan relasi (*relationship set*). (Abdul Kadir,2009,13)

Struktur logis (skema database) dapat ditunjukkan secara grafis dengan diagram ER yang dibentuk dari komponen-komponen berikut:

Tabel II.2 Notasi ERD (Entity Relationship Diagram)

Notasi	Keterangan
 Entitas	Persegi panjang mewakili kumpulan entitas
 Atribut	Elips mewakili atribut
 Relasi	Belah ketupat mewakili relasi
	Garis menghubungkan atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi

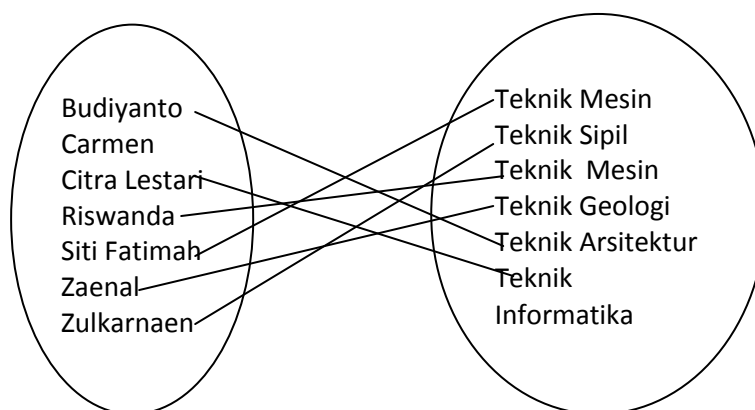
Sumber : Janner Simarmata & Imam Prayudi(2006 :59)

II.10.2. Jenis Hubungan

Terdapat beberapa jenis hubungan dalam ERD yaitu:

1. Hubungan one-to-one (1:1)

Menyatakan bahwa setiap entitas hanya berhubungan dengan satu entitas lainnya. Perhatikan contoh berikut



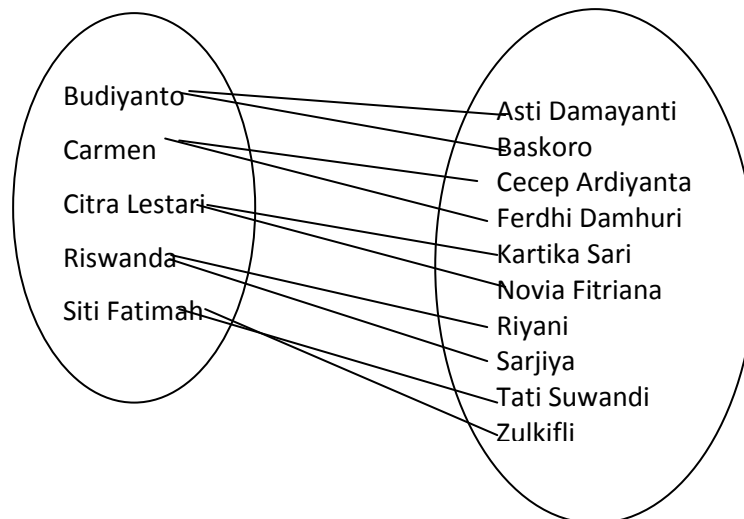
Gambar II.11. Contoh Hubungan One-To-One

Sumber : Abdul Kadir (2009:48)

Pada contoh tersebut terlihat bahwa setiap program studi hanya dikepalai oleh seorang dosen dan seorang dosen hanya bisa menjadi ketua sebuah program studi.

2. Hubungan one-to-many atau many-to-one

Menyatakan bahwa salah satu dari setiap entitas dapat berhubungan dengan lebih dari satu entitas lain. Hubungan one-to-many dan many-to-one sebenarnya sama saja. Yang membedakan adalah titik pandang terhadap tipe entitas yang menjadi acuan dalam hubungan. Perhatikan contoh berikut



Gambar II.12. Contoh Hubungan Many-To-One

Sumber : Abdul Kadir (2009:48)

Pada Gambar II.12. tampak bahwa:

1. Setiap dosen bisa membimbing lebih dari satu mahasiswa, tetapi bisa juga seorang dosen tidak menjadi pembimbing akademis sama sekali.

2. Setiap mahasiswa hanya memiliki satu pembimbing akademis.

3. Hubungan many-to-many (M:N)

Menyatakan bahwa antara entitas yang satu dapat berpasangan dengan banyak entitas begitu juga sebaliknya. Adapun contoh sebagai berikut:



Gambar II.13. Contoh Hubungan Many-To-Many

Sumber : Abdul Kadir (2009:48)

II.11. Normalisasi

Normalisasi adalah Teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi adalah proses langkah-langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel relational. Fungsi utama dari normalisasi ini adalah untuk membuat relasi yang terstruktur dengan baik yang bebas dari anomali-anomali

(ketidakkonsistenan yang terjadi saat penghapusan(*deleting*), pembaruan (*updating*), serta penambahan data (*inserting*).(Adi Nugroho,2009,44). Normalisasi sesungguhnya dapat dipahami sebagai tahapan-tahapan yang masing-masing berhubungan dengan bentuk normal dimana keadaan tabel yang dihasilkan dengan menerapkan aturan sederhana berkaitan dengan konsep ketergantungan fungsional (*functional dependency*) pada relasi yang bersangkutan. Berikut ada lima bentuk normal yang telah ditemukan yaitu:

1. Bentuk Normal Pertama (1NF/First Normal Form), adalah suatu bentuk relasi dimana atribut bernilai banyak (*multivlues atribut*) telah dihilangkan sehingga akan dijumpai nilai tunggal pada perpotongan setiap baris dan kolom satu nilai untuk irisan baris dan kolom pada tabel.
2. Bentuk Normal Kedua (2NF/Second Normal Form), suatu bentuk yang menyaratkan bahwa relasi harus sudah berada dalam bentuk normal pertama dan tidak mengandung dependensi parsial.
3. Bentuk Normal Ketiga (3NF/Third Normal Form), suatu keadaan yang menyaratkan bahwa relasi harus sudah berada dalam bentuk normal kedua dan tidak mengandung dependensi transitif (*transitive dependency*) telah dihilangkan.
4. Boyce-Codd Normal Form (BCNF), suatu keadaan yang menyaratkan bahwa setiap determinan (penentu) dalam suatu relasi berkedudukan sebagai kunci kandidat .
5. Bentuk Normal Keempat (4NF/Fifth Normal Form), suatu keadaan yang membuat relasi yang telah memenuhi bentuk normal keempat tidak dapat

didekomposisi menjadi relasi-relasi yang lebih kecil dengan kunci kandidat relasi-relasi pecahannyatersebut tidak sama dengan kunci kandidat relasi.
(Abdul Kadir:2009:130)

II.12. Kamus Data

Kamus data (KD) atau data dictionary atau disebut juga dengan istilah sistem data dictionary adalah catalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan KD, analisis sistem yang mendefinisikan data yang mengalir di sistem dengan lengkap. KD dibuat dengan tahap analisa dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem.(octaviani HS : 2010 :30).